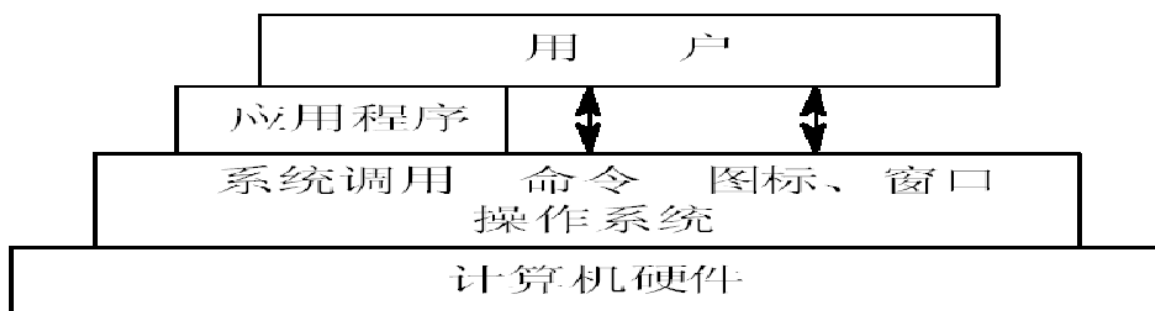


第一章 操作系统引论

1、操作系统的作用

也就是OS是什么

OS作为用户与计算机硬件系统之间的接口



OS作为计算机系统资源的管理者

OS实现了对计算机资源的抽象

2、操作系统的发展过程

首先是人工操作方式

然后是脱机输入/输出方式

批处理操作系统（单道多道）

分时系统

实时系统

3、三种基本操作系统的特征

批处理操作系统（单道、多道）

分时系统

实时系统

（列出一段说明，要知道是什么系统）

4、操作系统的基本特性

并发

共享

虚拟

异步

主要功能（后面就围绕这几个功能展开的）：

处理机管理

存储管理

设备管理

文件管理

用户接口

一些点

批处理操作系统（单道、多道），分时系统，实时系统

（列出一段说明，要知道是什么系统）

计算机系统中配置操作系统的目的是提高计算机的**利用率**和方便用户使用。

为了方便用户使用操作系统，利用操作系统所提供的各种功能和服务，操作系统通常向用户提供（命令）、（程序）、（图形）三种类型的接口。

在多道程序环境下，系统通常无法同时满足所有作业的资源要求，为使多道程序能有条不紊地运行，则操作系统应具有（处理机管理）、（存储器管理）、（设备管理）、（文件管理）管理模块，以实现对资源的管理。

操作系统定义：（是计算机系统中的一个系统软件,是一些程序模块的集合）

（多道程序）、（分时系统）的出现，标志操作系统的正式形成。

操作系统是加在（裸机）上加载的第一层软件，是对计算机硬件系统的（首次）扩充。

若一个操作系统具有很强的交互性，可同时供多个用户使用，则是（分时）操作系统。

若一个操作系统在用户提交作业后，不提供交互能力，只追求计算机资源的利用率、大吞吐量和作业流程的自动化则属于（批处理）操作系统。

所谓系统调用，就是用户在程序中（操作系统）所提供的一些子功能。

操作系统的四个特性，并发性 共享性 虚拟性 异步性

第二章 进程的描述与控制

在多道程序设计环境下，程序并发执行会出现

间断性：并发执行的程序之间相互制约导致“执行——暂停——执行”

失去封闭性：共享的资源由多个程序来改变。

不可再现性：

主要是失去封闭性和不可再现性，由此我们引入进程

1、进程的定义和特征

进程是程序在一个数据集合上的运行过程，是系统进行资源分配和调度的一个独立的基本单位。

引入了PCB进程控制块

结构特征：程序段、数据段、PCB

容易考填空（进程实体，也叫进程映像，就是由程序段、相关的数据段和PCB三部分构成的）

动态性：进程是程序的一次运行过程，有生命周期。

并发性

独立性：进程是一个能独立运行的基本单位，同时也是系统分配资源和调度的独立单位

异步性：进程按各自独立的、不可预知的速度向前推进

2、进程的基本状态及转换

三种状态

就绪状态（Ready）：得到了除CPU以外的所有必要资源。

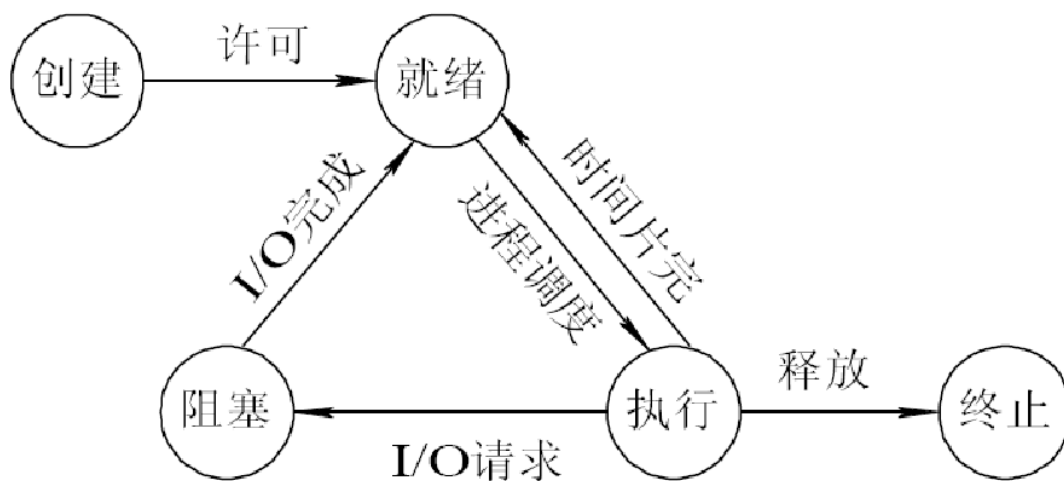
执行状态（Running）：已获得处理机，程序正在被执行。

阻塞状态（Blocked）：因等待某事件发生而暂时无法继续执行，从而放弃处理机，使程序执行处于暂停状态。

创建状态：进程已拥有了自己的PCB，但进程自身还未进入主存，即创建工作尚未完成，进程还不能被调度运行，其所处的状态。

终止状态：当一个进程到达了自然结束点，或是出现了无法克服的错误，或是被操作系统所终结，或是被其他有终止权的进程所终结，它将进入终止状态。

它们之间的转换



进程的五种基本状态及转换

3、进程控制块

作用 (P44)

概念

进程控制块是进程实体的重要组成部分，是操作系统中最重要
的记录型数据，在进程控制块PCB (Program Contral Block) 中记
录了操作系统所需要的、用于描述进程情况及控制进程运行所需要
的全部信息

作用

通过PCB，使得原来不能独立运行的程序（数据），成为一个
可以独立运行的基本单位，一个能够并发执行的进程。进程控制块
是进程存在的唯一标志。

进程控制块中的信息

进程标识符：唯一地标识一个进程。

内部标识符。在所有的操作系统中，都为每一个进程赋予了一个惟
一的数字标识符，它通常是一个进程的序号。

外部标识符。它由创建者提供，通常是由字母、数字组成，往往是
由用户(进程)在访问该进程时使用。

为了描述进程的家族关系，还应设置父进程标识及子进程标识。
还可设置用户标识，以指示拥有该进程的用户。

处理机状态：保留进程存放在处理器中的各种信息，主要由处理器内的各个寄存器的内容组成。

进程调度信息：进程状态、进程优先级、阻塞原因等等。

进程控制信息：程序和数据的地址、进程同步和通信机制、资源清单、链接指针

4、内核、原语、进程的创建

内核：加在硬件上的第一层软件，通过执行各种原语操作来实现各种控制和管理功能，具有创建、撤消、进程通信、资源管理的功能。

内核的基本功能

支撑功能：中断处理、时钟管理、原语操作

资源管理功能：进程管理、存贮管理、设备管理

原语：是由若干条机器指令所构成，用以完成特定功能的一段程序。
原语在执行期间是不可分割的或不可中断的。

创建原语、撤消原语、阻塞原语、唤醒原语

进程创建的过程

(进程创建原语Create())

申请一个空白PCB

为新进程分配资源

初始化PCB

新进程加入到就绪队列

5、临界资源、临界区

临界资源(Critical Resouce): 一次仅允许一个进程使用的资源, 如打印机、变量。

临界区是每个进程中访问临界资源的那段代码 (名词解释)

若能保证诸进程互斥地进入自己的临界区, 便可实现诸进程对临界资源的互斥访问。

6、同步机制应遵循的规则

- (1) 空闲让进。
- (2) 忙则等待。
- (3) 有限等待。
- (4) 让权等待

7、记录型信号量、利用记录型信号量解决生产者-消费者问题

信号量

由Dijkstra提出的一种解决进程的同步与互斥的工具-信号量 (semaphores) 。

信号量 - 用于表示资源数目或请求使用某一资源的进程个数的整型量

S是与临界区内所使用的公用资源有关的信号量

$S \geq 0$ 可供并发进程使用的资源数

$S < 0$ 正在等待使用临界区的进程数

整型信号量

S仅能通过两个标准的原子操作(Atomic Operation) wait(S)和 signal(S)来访问。这两个操作一直被分别称为P、V操作。 wait和 signal操作可描述为:

wait(S) {	signal(S)
while(S≤0);	{
S--;	S++;
}	}

记录型信号量

wait(S)和signal(S)操作可描述为:

wait(semaphore S)

```
{
    S.value--;
    if(S.value<0)    block(S.L)
}
```

signal(semaphore S)

```
{
    S.value++;
    if(S.value≤0)    wakeup(S.L);
}
```

经典进程同步问题；生产者与消费者问题

8、线程的基本概念

线程的引入

60年代，在OS中能拥有资源和独立运行的基本单位——进程
进程的创建、撤消与切换存在较大的时空开销——轻型进程
作为调度和分派的基本单位——线程（Threads）

线程与进程的比较

调 度：进程不再是调度的基本单位。

并 发 性：进程之间可以并发，线程之间也可以并发执行。

拥有资源：线程几乎不占资源，同一进程的线程共享进程的资源。

独立性：同一进程中的不同线程之间的独立性要低很多。

系统开销：线程的创建、撤消与切换的系统开销小的多。

支持多处理机系统

一些点

容易考填空（进程实体，也叫进程映像，就是由程序段、相关的数据段和PCB三部分构成的）

临界区是每个进程中访问临界资源的那段代码（名词解释）

进程是程序在一个数据集合上的运行过程，是系统进行资源分配和调度的一个独立的基本单位。

进程实体，也叫进程映像，就是由程序段、相关的数据段和PCB三部分构成的

进程五大特征，**动态性，并发性，独立性，异步性，结构性**

进程控制块中的信息包括**进程标识符（内部，外部），处理机状态，进程调度信息，进程控制信息**

多道程序环境中，操作系统分配资源以**进程**为基本单位。

对进程的管理和控制使用**原语**。

在一个单CPU系统中，若有5个用户进程，则处于就绪状态的用户进程最多有（**4**）个，最少有（**0**）个。

同一系统中如果两个进程要用有界缓冲来传输数据，这两个进程之间有一种制约关系，这种关系称为**同步**。

在多道程序环境下，多进程之间存在的制约有（资源共享）、（相互合作）。

并发程序间的相互制约关系可分为(同步)和(互斥)两种关系。

所谓同步，是指(直接相互制约关系)。互斥则是另一种进程间的制约关系，它是(间接制约关系)

所谓临界资源是（一次仅允许一个进程使用的资源）。

所谓临界区是（在每个进程中访问临界资源的那段代码）。

信号量的物理意义是：当信号量值大于0时表示（可用资源的数目）；当信号量值小于0时表示（因请求该资源而被阻塞的进程数目）。

操作系统中，对信号量S的P原语操作定义中，使进程进入相应等待队列等待的条件是（ $S < 0$ ）。

若一个进程已经进入临界区，其它要进入临界区的进程必须（等待）。

设有K个进程共享一临界区，问：

(1)一次只允许一个进程进入临界区，则信号量的初值为（1），变化范围是（ $-(k-1) \sim 1$ ）。

(2)一次允许M($M < K$)个进程进入临界区，则信号量的初值为（M），变化范围是（ $-(k-m) \sim m$ ）。

线程在执行中需要资源，**不可以**直接向系统申请

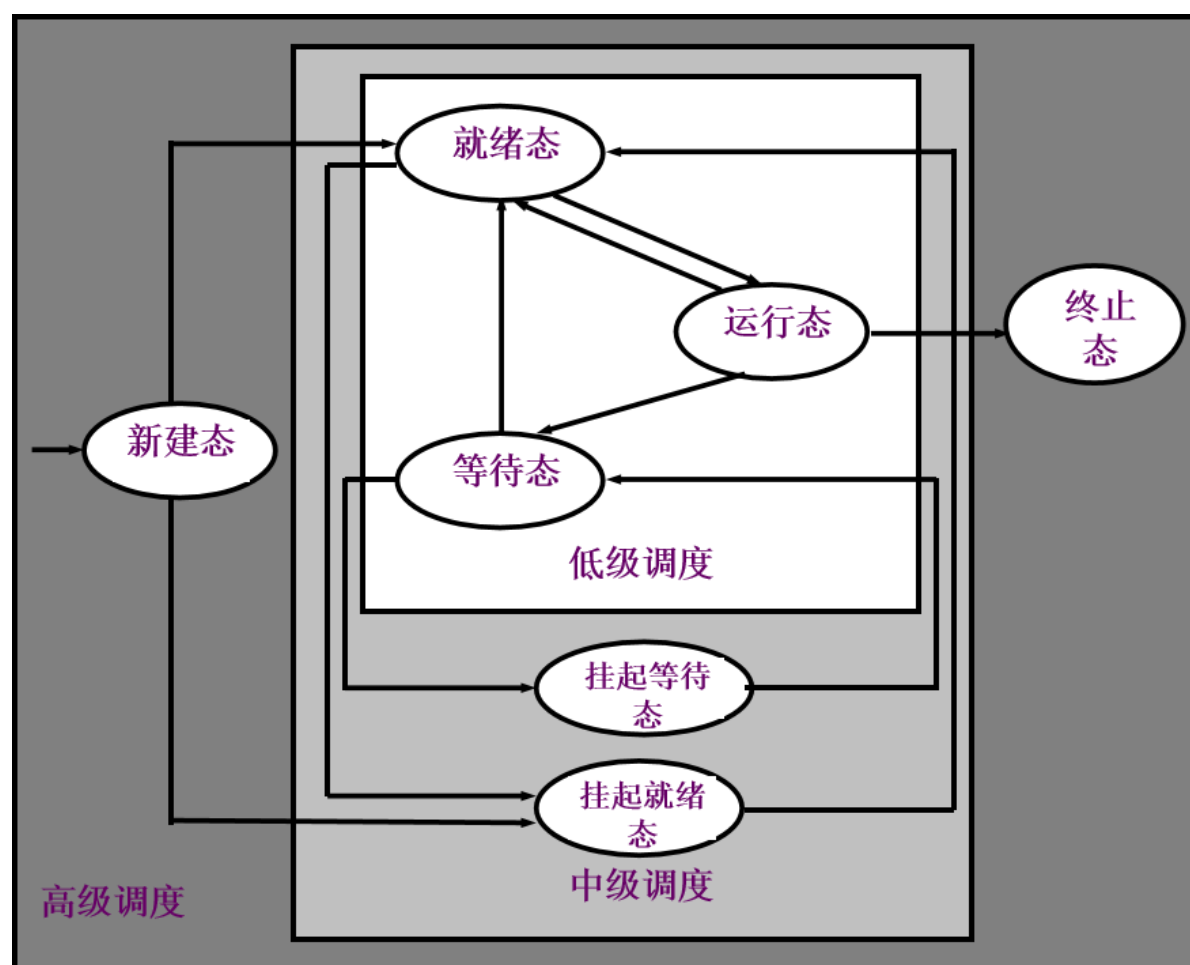
第三章 处理机调度与死锁

1、处理机调度的层次和调度算法的目标

高级调度（作业调度）

中级调度（内存调度）

低级调度（进程调度）



➤ 处理机调度算法的目标

➤ 共同目标

➤ 资源利用率

CPU的利用率=

CPU有效工作时间/（CPU有效工作时间+CPU空闲等待时间）

➤ 公平性

➤ 平衡性

➤ 策略强制执行

➤ 处理机调度算法的目标

➤ 批处理系统的目标

➤ 平均周转时间短

周转时间：从作业被提交给系统开始，到作业完成为止。

平均周转时间：

$$T = \frac{1}{n} \left[\sum_{i=1}^i T_i \right]$$

带权周转时间：作业的周转时间 T 与系统为它提供服务的时间 T_s 之比，即 $W=T/T_s$ 。

平均带权周转时间：

$$W = \frac{1}{n} \left[\sum_{i=1}^n \frac{T_i}{T_{Si}} \right]$$

选择调度算法的准则，周转时间，带权周转时间，响应时间是什么，如何计算？（猜测要考）

2、作业调度的三种算法：先来先服务、短作业优先和高优先权优先

先来先服务调度算法

- 可用于作业调度和进程调度。

- 按照作业（进程）进入后备队列（就绪队列）的先后次序来挑选作业（进程），先进入先被挑选。
- 算法容易实现，效率不高，只顾及作业（进程）等候时间，没考虑作业（进程）要求服务时间的长短。
- 有利于长作业（进程），不利于短作业（进程）。有利于CPU繁忙型作业，不利于I/O繁忙型作业。

进程名	到达时间	服务时间	开始执行时间	完成时间	周转时间	带权周转时间
A	0	1	0	1	1	1
B	1	100	1	101	100	1
C	2	1	101	102	100	100
D	3	100	102	202	199	1.99

短作业(进程)优先调度算法

短作业优先（SJF）

从后备队列中选择估计运行时间最短的作业，调入内存运行。

短进程优先（SPF）

从就绪队列中选出估计运行时间最短的进程，将处理机分配给它，使它立即执行。

直到运行完成进程才会让出处理机--非抢占式。

调度算法 \ 作业情况	进程名	A	B	C	D	E	平均
	到达时间	0	1	2	3	4	
	服务时间	4	3	5	2	4	
FCFS (a)	完成时间	4	7	12	14	18	
	周转时间	4	6	10	11	14	9
	带权周转时间	1	2	2	5.5	3.5	2.8
SJF (b)	完成时间	4	9	18	6	13	
	周转时间	4	8	16	3	9	8
	带权周转时间	1	2.67	3.1	1.5	2.25	2.1

高优先级优先调度算法

作为作业调度算法时，系统将从后备队列中选择若干个优先级最高的作业装入内存

作为进程调度算法时，该算法把处理机分配给就绪队列中优先级最高的进程

非抢占式优先级算法

抢占式优先级调度算法

3、进程调度

进程调度的方式

非抢占方式

一旦把处理机分配给某个进程后，让该进程一直执行，直到该进程完成或者发生某事件而阻塞。

引起进程调度的因素：

正在执行的进程执行完毕

执行中的进程因为提出I/O请求而暂停执行

进程通信或同步过程中执行了原语操作

抢占方式

允许调度程序根据某种原则，暂停正在执行的进程，将处理机重新分配给其他进程。

抢占原则：

优先级原则：允许优先级高的新到进程抢占当前进程的处理机

短进程优先原则

时间片原则

4、死锁的定义、产生死锁的必要条件、死锁的原因

死锁的定义

如果一组进程中的每一个进程都在等待仅由该组进程中的其他进程才能引发的事件，那么该组进程是死锁的（deadlock）。

产生死锁的必要条件（简答）

互斥条件：进程互斥使用资源

请求和保持条件：申请新资源时不释放已占有资源

不剥夺条件：一个进程不能抢夺其他进程占有的资源

环路等待条件：存在一组进程循环等待资源的环形链

处理死锁的基本方法

预防死锁：通过设置某些限制条件来破坏产生死锁的四个必要条件中的一个或几个，来预防发生死锁

避免死锁：在动态分配资源的过程中，用某种方法防止系统进入不安全状态，从而避免发生死锁

检测死锁：通过设置检测机制，及时检测出死锁的发生，确定有关的进程和资源

解除死锁：与检测死锁配套使用，常用的方法是撤销或挂起一些进程，收回资源，分配给处于阻塞状态的进程，使之转为就绪状态，可以继续运行

5、预防死锁

（静态资源分配法等等，看一下）

p116

6、避免死锁：安全状态，银行家算法

系统安全状态

- 安全状态是指系统能按某种顺序为每个进程分配其所需的资源，直至满足每个进程对资源的最大需求，使每个进程都可顺利地完

安全序列：进程安全执行完的顺序。（怎么找？）

如果系统无法找到这样一个安全序列，则称系统处于不安全状态。

利用银行家算法避免死锁（必考，容易考大题前面的进程调度算法也容易考大题）

看书，就是给了资源后无法找到安全序列就不允许分配资源，反之允许

P121

一些点

选择调度算法的准则，周转时间，带权周转时间，响应时间是什么，如何计算？（猜测要考）

产生死锁的必要条件（简答）

安全序列：进程安全执行完的顺序。（怎么找？）

利用银行家算法避免死锁（必考，容易考大题前面的进程调度算法也容易考大题）

第四章 存储器管理

1、三种装入方式

程序的装入

绝对装入方式

可重定位装入方式

- 重定位：在装入时对目标程序中指令和数据地址的修改过程称。或者说：把作业地址空间中使用的逻辑地址变换成内存空间中的物理地址的过程。又称地址映射。
- 地址变换在装入时一次完成，以后不再改变，故称为静态重定位。

动态运行时装入方式

装入时动态链接方式

- 采用边装入边链接的方式

运行时动态链接方式

2、三种链接方式

静态链接方式

装入时动态链接方式

- 采用边装入边链接的方式

运行时动态链接方式

3、连续分配存储管理方式、紧凑

单一连续分配

固定分区分配

动态分区分配

对比理解，比如注意一下有没有内部碎片之类的

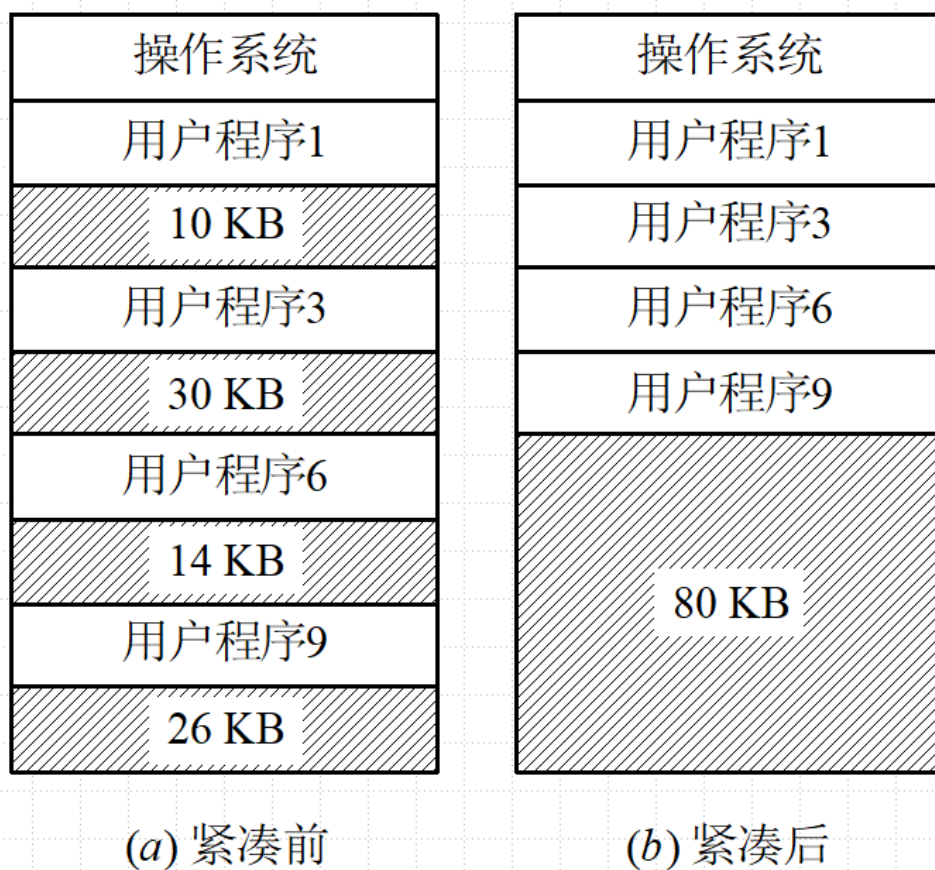
动态重定位的引入

碎片问题

经过一段时间的分配回收后，内存中存在很多很小的空闲块。它们每一个都很小，不足以满足分配要求；但其总和满足分配要求。这些空闲块被称为“碎片”或“零头”。

碎片问题的解决——“拼接”或“紧凑”

移动内存中作业的位置，以把原来多个分散的小分区拼接成一个大分区。



动态重定位的实现

每次“紧凑”后必须对移动了的程序或数据进行重定位

地址变换过程是在程序执行期间，随着对每条指令或数据的访问自动进行的，称为动态重定位。

硬件地址变换机构的支持——重定位寄存器

内存地址 = 相对地址 + 重定位寄存器中的地址

移动程序时只需修改重定位寄存器中的地址

4、对换的概念

所谓“对换”，是指把内存中暂时不能运行的进程或者暂时不用的程序和数据，调出到外存上，以便腾出足够的内存空间，再把已具备运行条件的进程或进程所需要的程序和数据换入内存。

对换分类

- 整体对换、进程对换
- 部分对换：页面对换、分段对换

5、分页存储管理方式：页、物理块、地址变换、快表

分页的地址变换（比如问每页大小，会计算，块号，页，等等，必考）

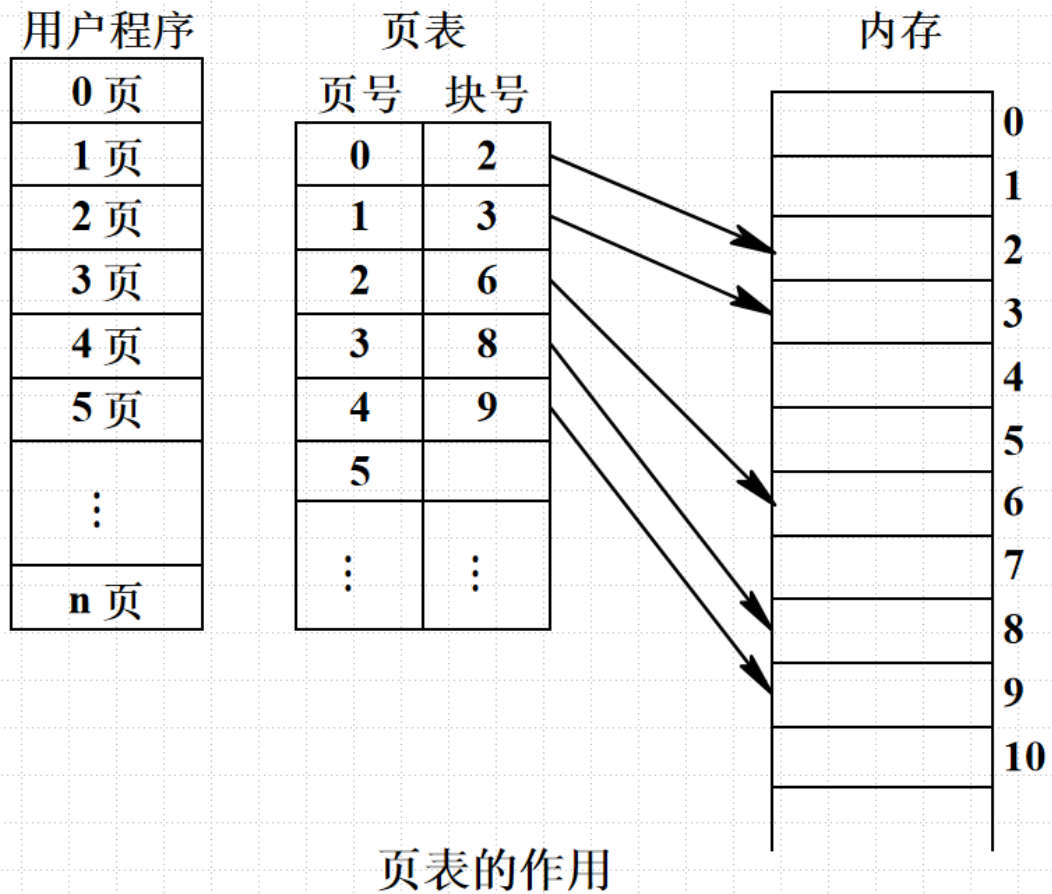
页（页面）——把每个作业(进程)逻辑地址空间划分成若干大小相等的片.第0页、第1页

（物理）块或页框(frame)——把内存空间分成与页面相同大小的若干个存储块。0 # 块、1 # 块

页内碎片——由于进程的最后一页经常装不满一块而形成了不可利用的碎片。

页表（页面映像表）

页表的作用是实现从页号到物理块号的地址映射



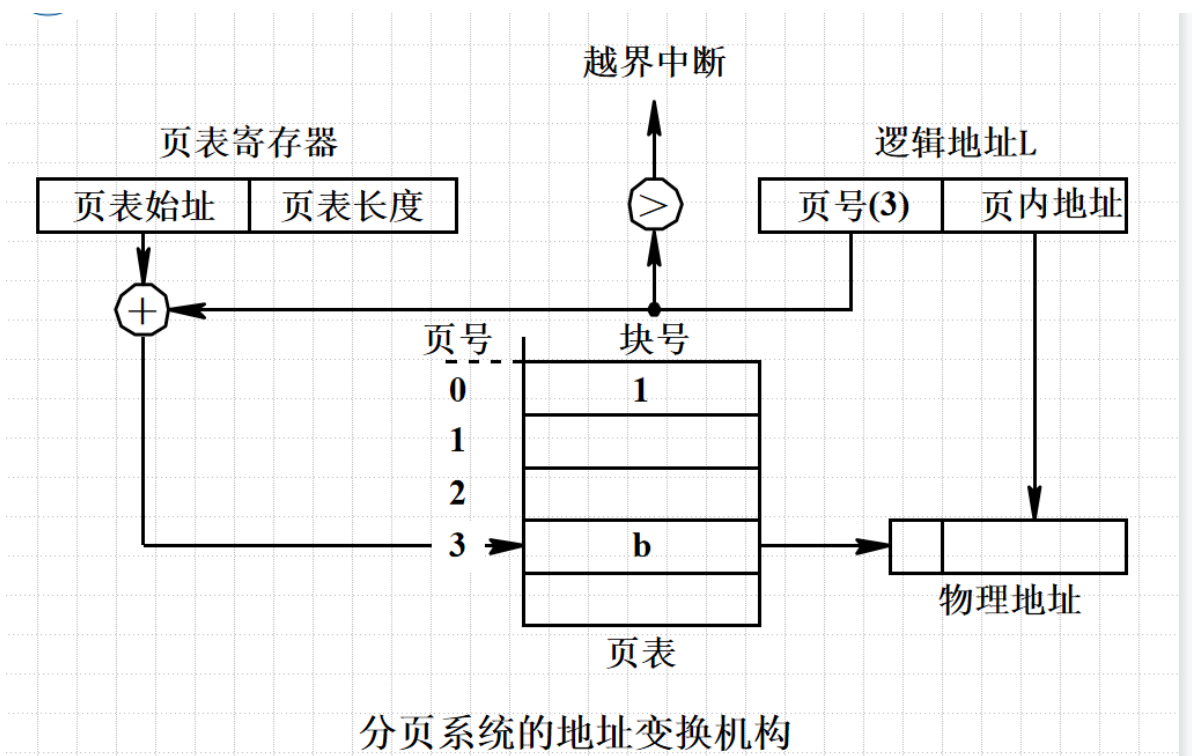
借助于页表，实现从逻辑地址到物理地址的转换

基本的地址变换机构

页表驻留在内存中

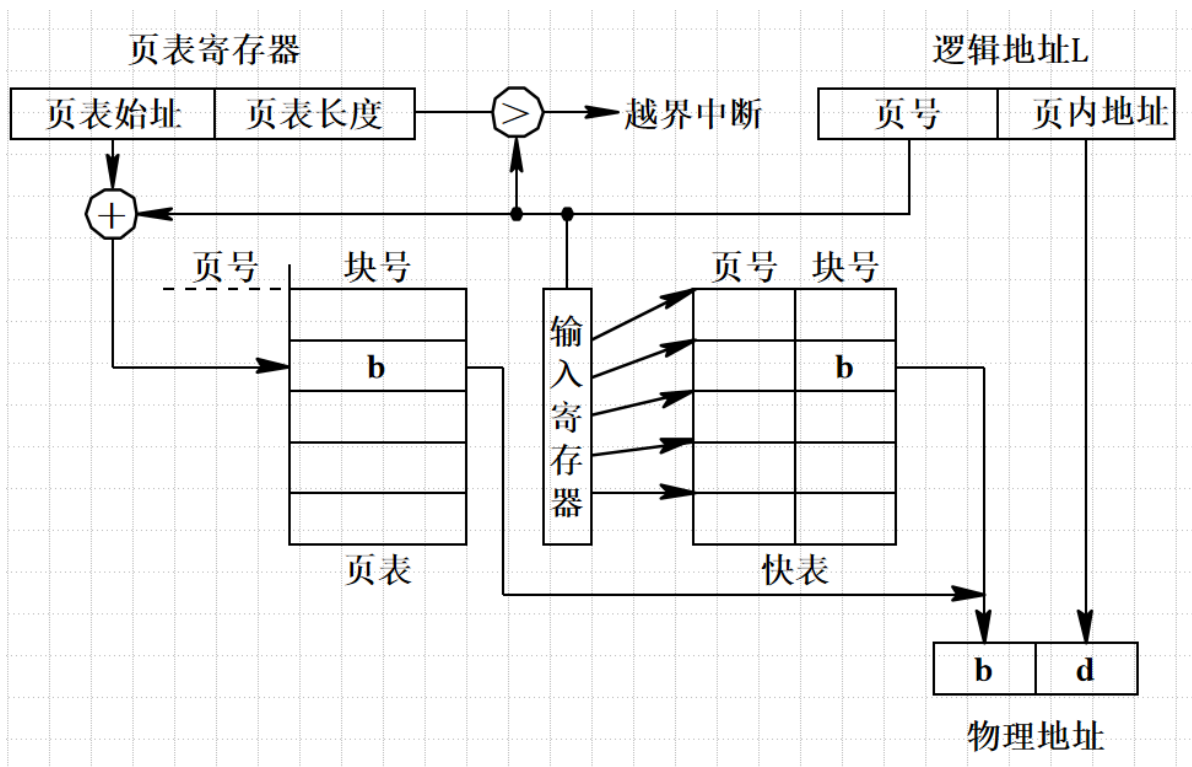
进程未执行时，页表的始址和页表长度存放在PCB中

进程执行时，页表的始址和页表长度装入页表寄存器



联想寄存器（快表）：存放当前访问的若干个页表项的一个高速缓冲寄存器（名词解释）

理解块表的作用



6、分段存储管理方式：分页与分段的比较

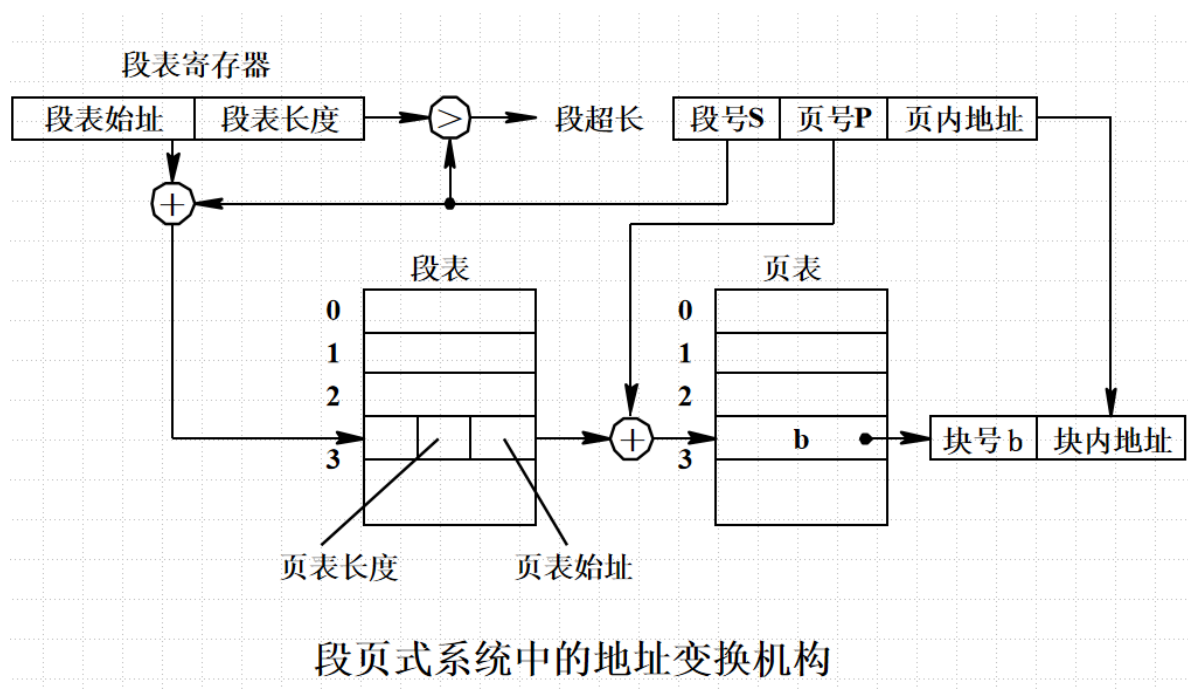
分页和分段的主要区别

- (1) 页是信息的物理单位，由于系统管理的需要。段则是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能够更好地满足用户的需要。
- (2) 页的大小固定且由系统决定，而段的长度却不固定，它由其完成的功能决定。
- (3) 分页的作业地址空间是一维的而分段的作业地址空间则是二维的，程序员在标识一个地址时，需给出段名和段内地址。
- (4) 由于段是信息的逻辑单位，因此便于存贮保护和信息的共享，页的保护和共享受到限制。

7、段页式存储管理方式

原理

先将程序分成若干个段，再把每个段分成若干个页，并为每一个段赋予一个段名。



一些点

单一连续分配

固定分区分配

动态分区分配

对比理解，比如注意一下有没有内部碎片之类的

分页的地址变换（比如问每页大小，会计算，块号，页，等等，必考）

联想寄存器（快表）：存放当前访问的若干个页表项的一个高速缓冲寄存器（名词解释）

第五章 虚拟存储器

1、虚拟存储器的定义和特征

为啥引入？常规存储器是怎样的

虚拟存储器定义

所谓虚拟存储器，是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。（简答）

特征（填空）

离散性

多次性

对换性

虚拟性

2、请求分页存储管理：页表、缺页中断、地址变换

页表机制（每一项的作用，自己看书，重要！！）

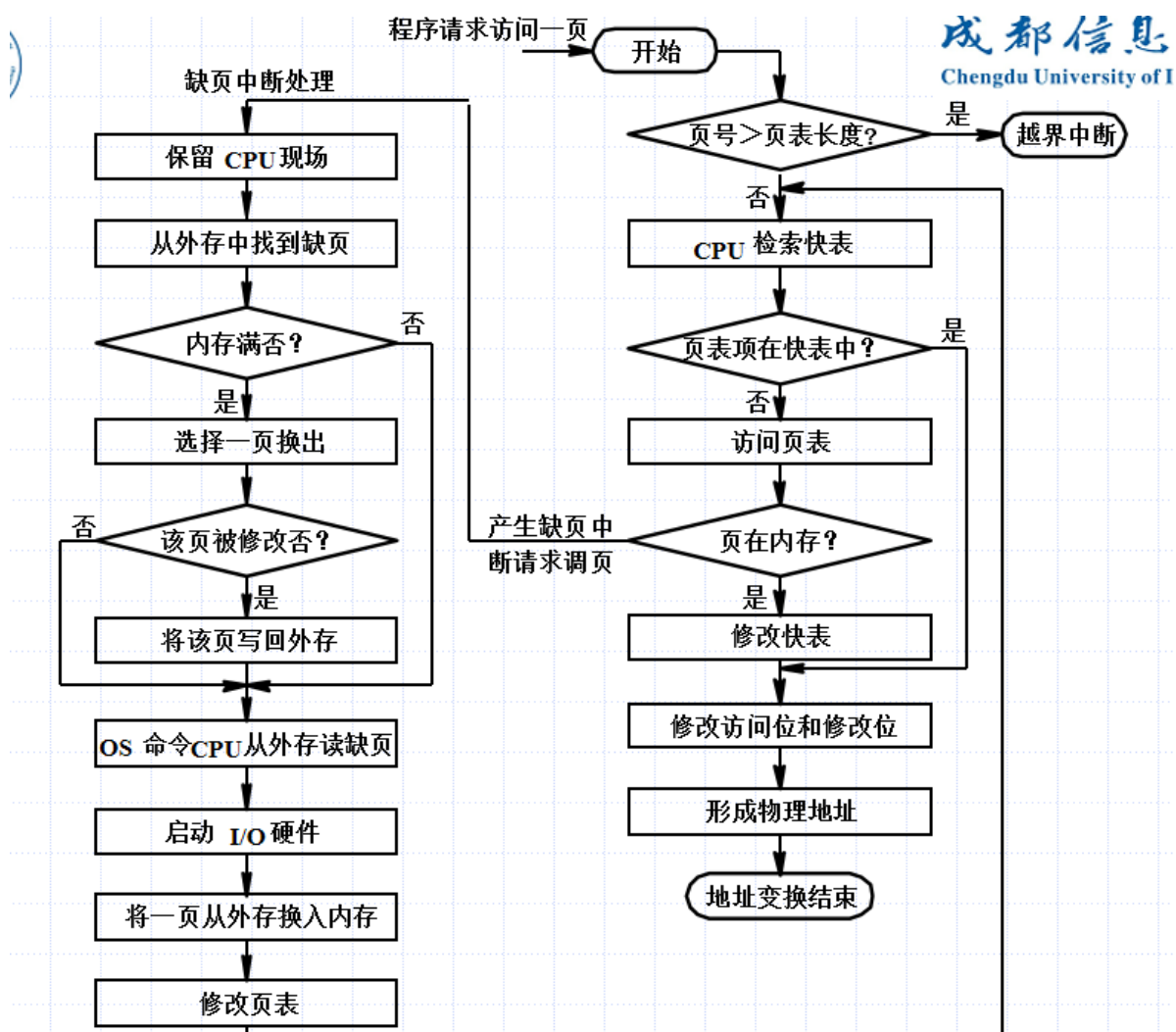
页号	物理块号	状态位P	访问字段A	修改位M	外存地址
----	------	------	-------	------	------

缺页中断机构

当要访问的页面不在内存时，便产生一缺页中断，请求OS将所缺的页面调入内存。经历保护CPU环境、分析中断原因、转入缺页中断处理程序进行处理、恢复CPU环境等步骤。

特点：在指令执行期间产生和处理中断信号；一条指令在执行期间，可能产生多次缺页中断。

地址变换机构



3、页面置换算法：最佳置换算法\先进先出置换算法\最近最久未使用(LRU) 置换算法

一定要掌握！！（不难，细心就行）

➤ 最佳置换算法

引用率

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		2			2			2				7		
	0	0	0		0		4			0			0			0	0		
		1	1		3		3			3			1			1	1		

页框(物理块)

页面置换：**6**次

➤ 先进先出(FIFO)页面置换算法

引用率

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2	2	4	4	4	0			0	0			7	7	7
	0	0	0		3	3	3	2	2	2			1	1			1	0	0
		1	1		1	0	0	0	3	3			3	2			2	2	1

页框

页面置换：**12**次

➤ 最近最久未使用 (LRU) 置换算法

引用率

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		

页框

页面置换：**9**次

4、进程的抖动

产生“抖动”的原因

运行的进程太多，每一个进程的物理块太少，频繁地出现缺页。使得在系统中排队等待页面调进/调出的进程数目增加。**对磁盘的有效访问时间也随之急剧增加，造成每个进程的大部分时间都用于页面的换进/换出，而几乎不能再去做任何有效的工作，从而导致发生处理机的利用率急剧下降并趋于0的情况。我们称此时的进程是处于“抖动”状态。**（简答，名词）

一些点

虚拟存储器定义

所谓虚拟存储器，是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。（简答）

特征（填空）

离散性

多次性

对换性

虚拟性

页面置换算法：最佳置换算法\先进先出置换算法\最近最久未使用（LRU）置换算法

一定要掌握！！（不难，细心就行）

产生“抖动”的原因

运行的进程太多，每一个进程的物理块太少，频繁地出现缺页。使得在系统中排队等待页面调进/调出的进程数目增加。**对磁盘的有效访问时间也随之急剧增加，造成每个进程的大部分时间都用于页面的换进/换出，而几乎不能再去做任何有效的工作，从而导致发生处理机的利用率急剧下降并趋于0的情况。我们称此时的进程是处于“抖动”状态。**（简答，名词）

第六章 输入输出系统

1、I/O系统的功能、模型

p195（名词解释吧）

什么是块设备

什么是字符设备

I/O系统的基本功能

隐藏物理设备的细节

仅向上层进程提供少量的、抽象的读/写命令

与设备的无关性

提高处理机和I/O设备的利用率

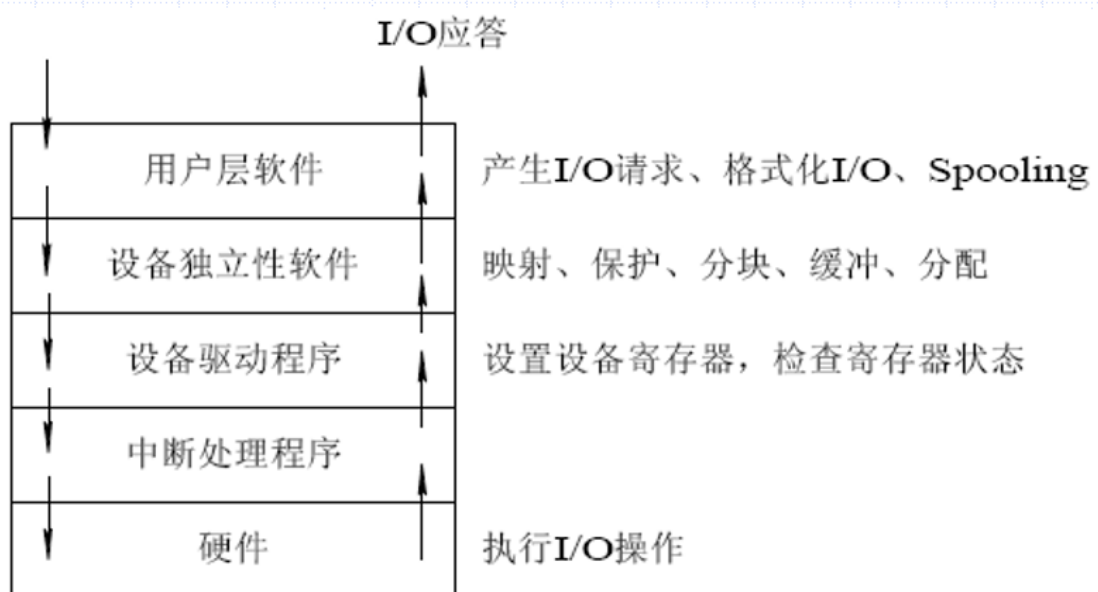
对I/O设备进行控制

确保对设备的正确共享

错误处理

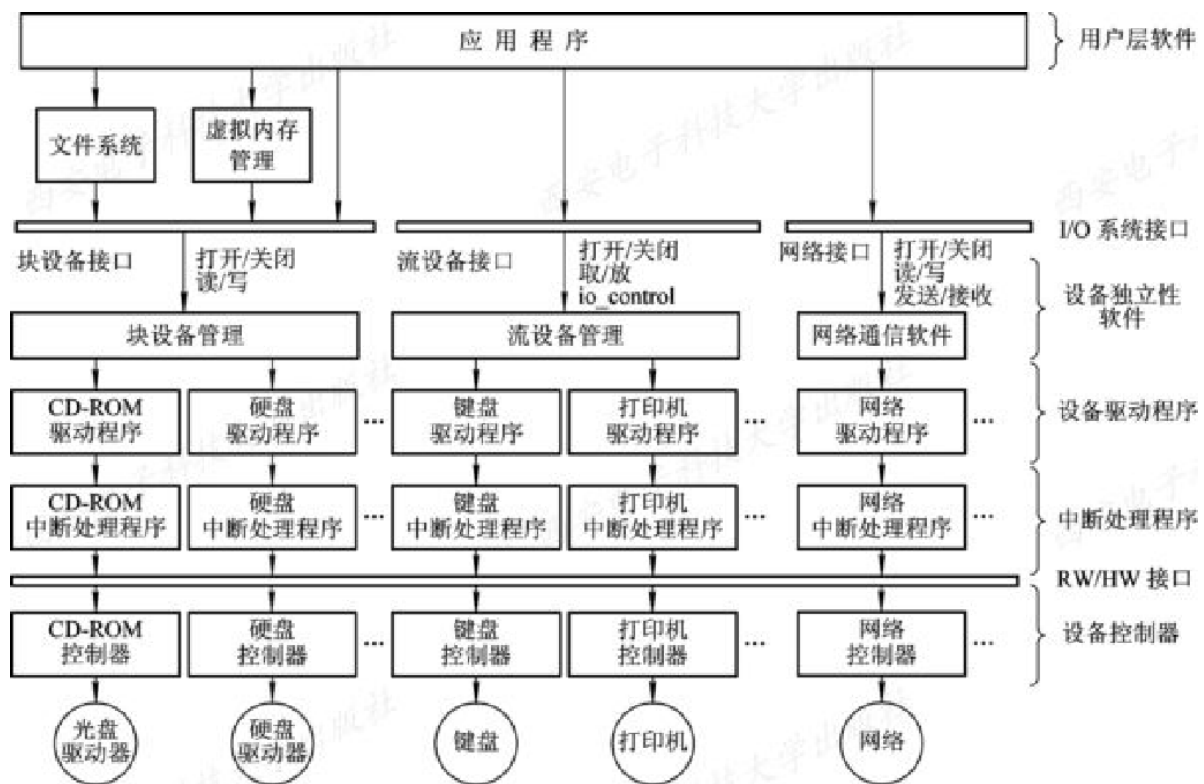
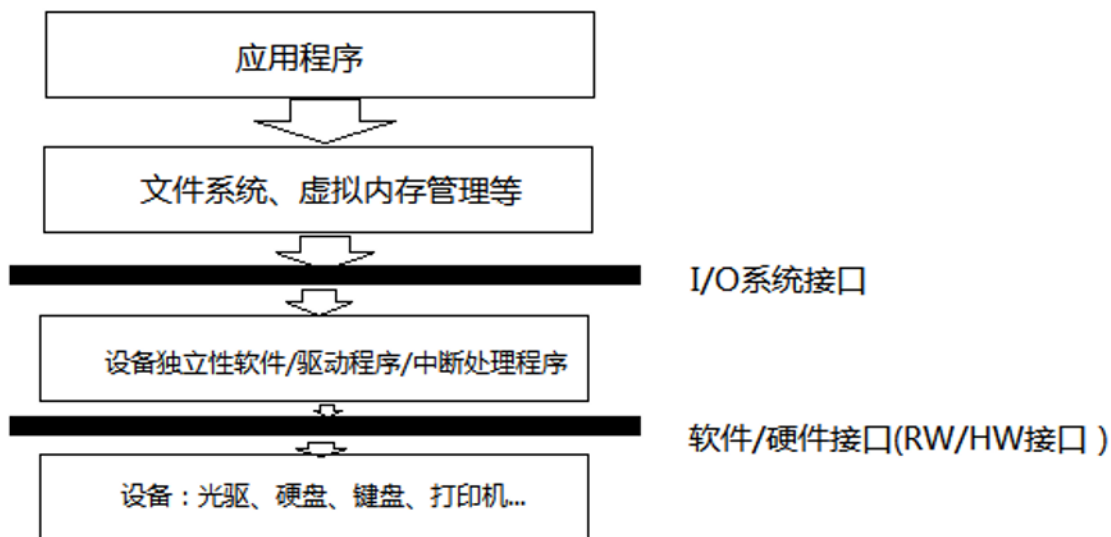
I/O系统的层次结构和模型

➤ I/O系统的层次结构



I/O 系统的层次及功能

➤ I/O系统中各种模块之间的层次视图



2、通道

I/O通道

定义：通道是独立于CPU的专门负责数据输入/输出传输工作的处理机，对外部设备实现统一管理，代替CPU对输入/输出操作进行控制，从而使输入，输出操作可与CPU并行操作。

引入通道的目的：为了使CPU从I/O事务中解脱出来，同时为了提高CPU与设备，设备与设备之间的并行工作能力

通过执行通道程序来控制I/O操作
指令类型单一，与CPU共享内存

3、I/O设备的控制方式

按照I/O控制器功能的强弱，以及和CPU之间联系方式的不同，对I/O设备的控制方式分类，
主要差别在于：中央处理器和外围设备并行工作的方式不同，并行工作的程度不同

轮询

中断

DMA（直接存储器访问）

这三个的区别

4、设备独立性（设备无关性）的定义，如何实现

也称为设备无关性。其基本含义是：应用程序独立于具体使用的物理设备。/应用程序中所用的设备，不局限于使用某个具体的物理设备。（名词）

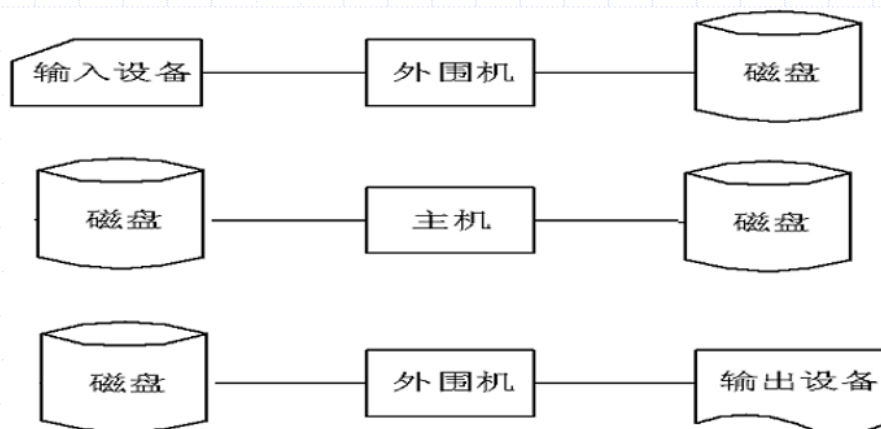
实现：引入了逻辑设备和物理设备这两个概念

在应用程序中，使用逻辑设备名称来请求使用某类设备；而系统在实际执行时，还必须使用物理设备名称。

系统须具有将逻辑设备名称转换为某物理设备名称的功能

5、假脱机系统的特点

► 什么是SPOOLing



多道程序环境中，用程序来模拟脱机输入时的外围控制机功能。

(简答?) 特点:

- 提高了I/O的速度。
- 将独占设备改造为共享设备。
- 实现了虚拟设备功能。

6、引入缓冲的目的

改善中央处理器与外围设备之间速度不配的矛盾，凡是数据到达和离去速度不匹配的地方均可采用缓冲技术。

减少对CPU的中断频率，放宽对CPU中断响应时间的限制，

解决数据粒度不匹配的问题

提高CPU和I/O设备的并行性。

7、磁盘调度算法

磁盘访问时间有哪些

寻道时间，旋转延迟时间，传输时间

目标

使磁盘的平均寻道时间最少

算法（一定要掌握）

先来先服务FCFS(First-Come, First Served)□

最短寻道时间优先SSTF(Shortest Seek Time First)

扫描(SCAN)算法

循环扫描(CSCAN)算法

一些点

p195（名词解释吧）

什么是块设备

什么是字符设备

也称为设备无关性。其基本含义是：应用程序独立于具体使用的物理设备。/应用程序中所用的设备，不局限于使用某个具体的物理设备。（名词）

spooling（简答？）特点：

- 提高了I/O的速度。
- 将独占设备改造为共享设备。
- 实现了虚拟设备功能。

算法（一定要掌握）

先来先服务FCFS(First-Come, First Served)□

最短寻道时间优先SSTF(Shortest Seek Time First)

扫描(SCAN)算法

循环扫描(CSCAN)算法

第七章 文件管理

1、文件分类

文件名和扩展名

文件类型（填空）

按文件的性质和用途分为：系统文件，库文件，用户文件

按文件中数据的形式分类：源文件，目标文件，可执行文件

按存取控制属性分类：只执行文件，只读文件，读写文件

按组织形式和处理方式：普通文件，目录文件，特殊文件

2、文件系统的层次结构

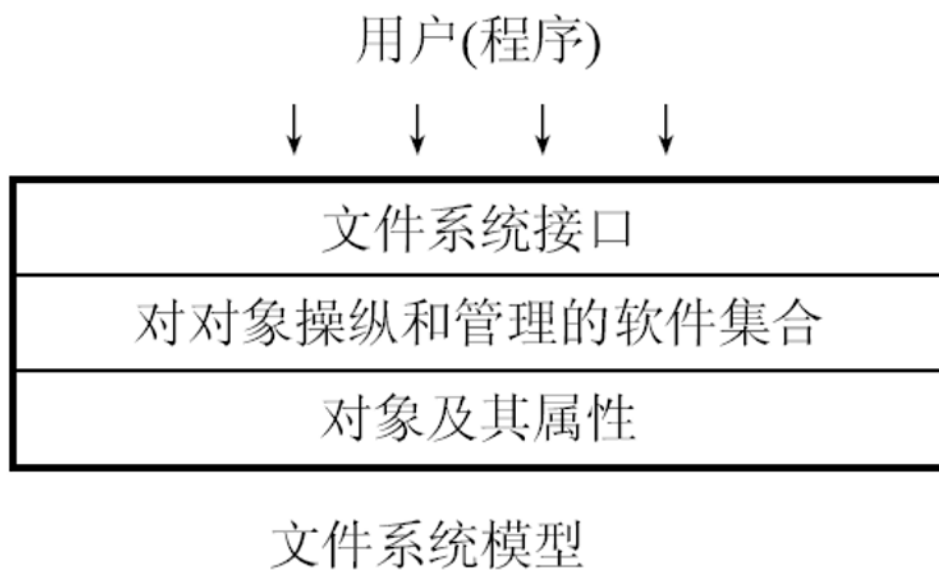
文件管理系统管理的对象有那些（简答，填空）

文件

目录

磁盘存储空间

► 文件系统的层次结构



3、文件的逻辑结构：概念及分类

文件的逻辑结构（文件的组织）：从用户角度看到的文件的全貌，也就是它的记录结构。

文件的物理结构（文件的存储结构）：文件在外存上的存储组织形式。（名词）

按是否有结构分类

有结构文件：定长记录，变长记录

无结构文件：流式文件

按文件的组织方式分类（填空）

顺序文件

索引文件（这个是逻辑地址）

索引顺序文件

4、对目录管理的要求

对目录管理的要求如下：（简答）

实现“按名存取”。

提高对目录的检索速度。

文件共享。

允许文件重名。

5、文件控制块与索引结点

➤ 文件控制块

➤ 基本信息类

- ① 文件名； ② 文件物理位置；
- ③ 文件逻辑结构； ④ 文件的物理结构

➤ 存取控制信息类

➤ 使用信息类

文 件 名	扩 展 名	属 性	备 用	时 间	日 期	第 一 块 号	盘 块 数
-------------	-------------	--------	--------	--------	--------	------------------	-------------

MS-DOS的文件控制块

索引结点不包含文件名 (! ! !)

➤ 索引结点

- 文件目录的缺陷
- 索引结点的引入
 - 目录：文件名和指向**I**结点的指针
 - **i** 结点：文件描述信息。

6、两种文件共享方法

基于索引结点的共享方式

利用符号链实现文件共享

一些点

文件类型（填空）

按文件的性质和用途分为：系统文件，库文件，用户文件

按文件中数据的形式分类：源文件，目标文件，可执行文件

按存取控制属性分类：只执行文件，只读文件，读写文件

按组织形式和处理方式：普通文件，目录文件，特殊文件

文件管理系统管理的对象有那些（简答，填空）

文件

目录

磁盘存储空间

文件的逻辑结构（文件的组织）：从用户角度看到的文件的全貌，也就是它的记录结构。

文件的物理结构（文件的存储结构）：文件在外存上的存储组织形式。（名词）

按是否有结构分类

有结构文件：定长记录，变长记录

无结构文件：流式文件

按文件的组织方式分类（填空）

顺序文件

索引文件（这个是逻辑地址）

索引顺序文件

对目录管理的要求如下：（简答）

实现“按名存取”。

提高对目录的检索速度。

文件共享。

允许文件重名。

索引结点不包含文件名（！！！！）

第八章 磁盘存储器的管理

1、文件的物理结构：概念及分类

由什么样的外存组织方式，就会有什么样的文件结构

P268

外存有连续组织方式，形成的文件是顺序式文件结构

链接组织方式，形成链接式（有显式链接，隐式链接）

索引组织，形成索引式

2、FAT（重点）和索引组织方式

FAT，文件分配表（名词）

作用

宽度，以及它会影响啥东西

会计算（272-273）

以上都是重点

在微软公司的早期MS-DOS中，所使用的是12 位的FAT12 文件系统，后来为16 位的

FAT16文件系统；在Windows 95和Windows 98操作系统中则升级为32位的FAT32；Windows

NT、Windows 2000 和Windows XP 操作系统又进一步发展为新技术文件系统NTFS(New

Technology File System)。上述的几种文件系统所采用的文件分配方式基本上都是类似于前

一节所介绍的显式链接方法。

FAT12

对于1.2 MB的软盘，每个盘块的大小为512 B，在每个FAT中共含有2.4 K 个表项，由于每个FAT表项占12 位，故FAT 表占用3.6 KB的存储空间。

$$1.2\text{MB}/512\text{B}=2.4\text{k}$$

$$2.4\text{K}\times 1.5=3.6\text{KB}$$

FAT12

每个FAT表项为12位，所以FAT表中最多允许4K表项，盘块（扇区）512B，每一个磁盘分区的容量为： $4\text{K}\times 512\text{B}=2\text{MB}$

一个物理磁盘能支持4个逻辑磁盘分区，所以磁盘最大容量为8MB

为了增加容量，引入族的概念：

一个族可以是512B、1KB 、2KB、4KB...

问题：族内碎片增加。

FAT16

FAT表的宽度增加到16位，长度增加到 2^{16}

如果族为4KB，分区容量为： $2^{16}\times 4\text{KB}$

FAT32

FAT表的宽度增加到32位，长度增加到 2^{32}

如果族为4KB，分区容量为： $2^{32}\times 4\text{KB}$

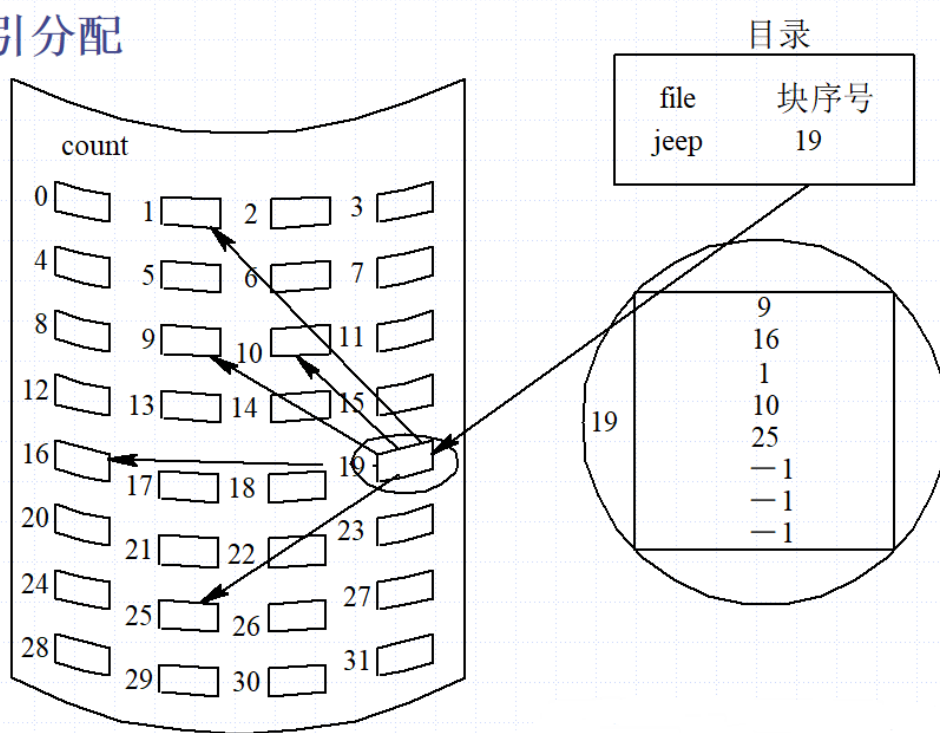
缺点：

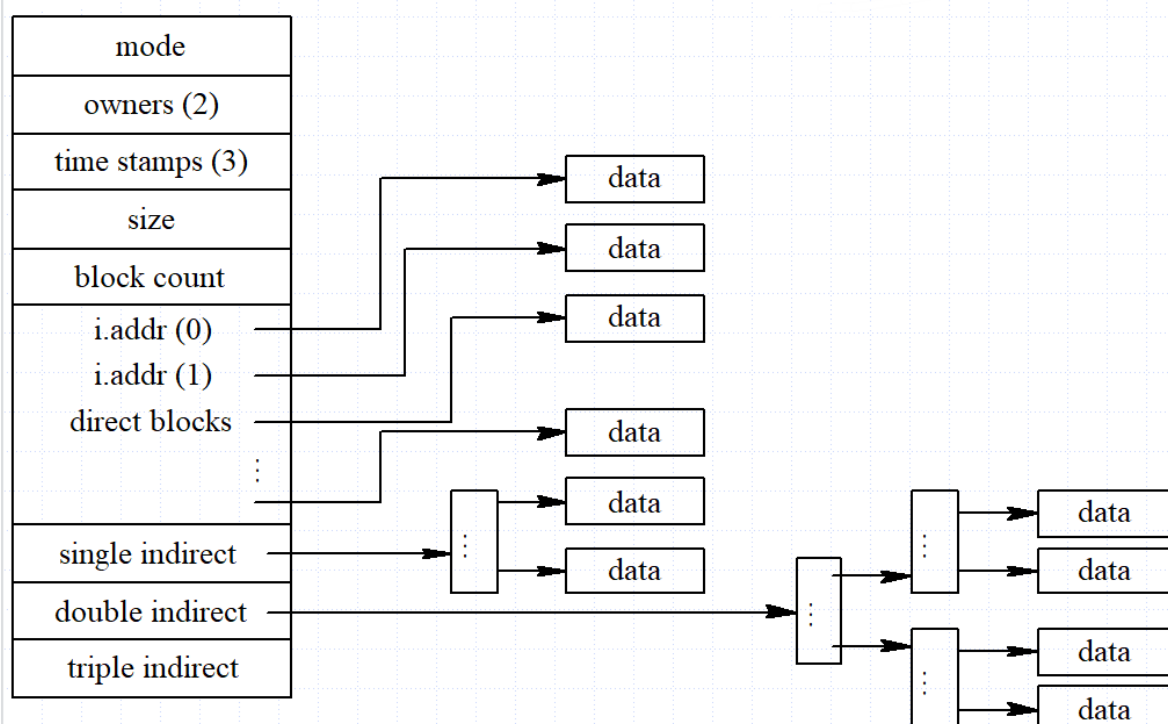
文件分配表大，运行速度慢；不支持容量小于512MB的分区（仍然需使用FAT16）；单个文件不能大于4GB；不保持向下兼容。

索引组织方式

索引组织方式

1. 单级索引分配





3、位示图法与成组链接法

位示图法（重点）

➤ 位示图法

1. 位示图

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
16																

2. 盘块的分配

(1) 顺序扫描位示图，从中找出一个或一组其值为“0”的二进制位(“0”表示空闲时)。

(2) 将所找到的一个或一组二进制位，转换成与之相应的盘块号。假定找到的其值为“0”的二进制位，位于位示的第*i*行、第*j*列，则其相应的盘块号应按下式计算：

$$b=n(i-1)+j$$

式中，*n*代表每行的位数。

(3) 修改位示图，令map [*i,j*] =1。

3. 盘块的回收

(1) 将回收盘块的盘块号转换成位示图中的行号和列号。转换公式为：

$$i=(b-1)\text{DIV } n+1$$

$$j=(b-1)\text{MOD } n+1$$

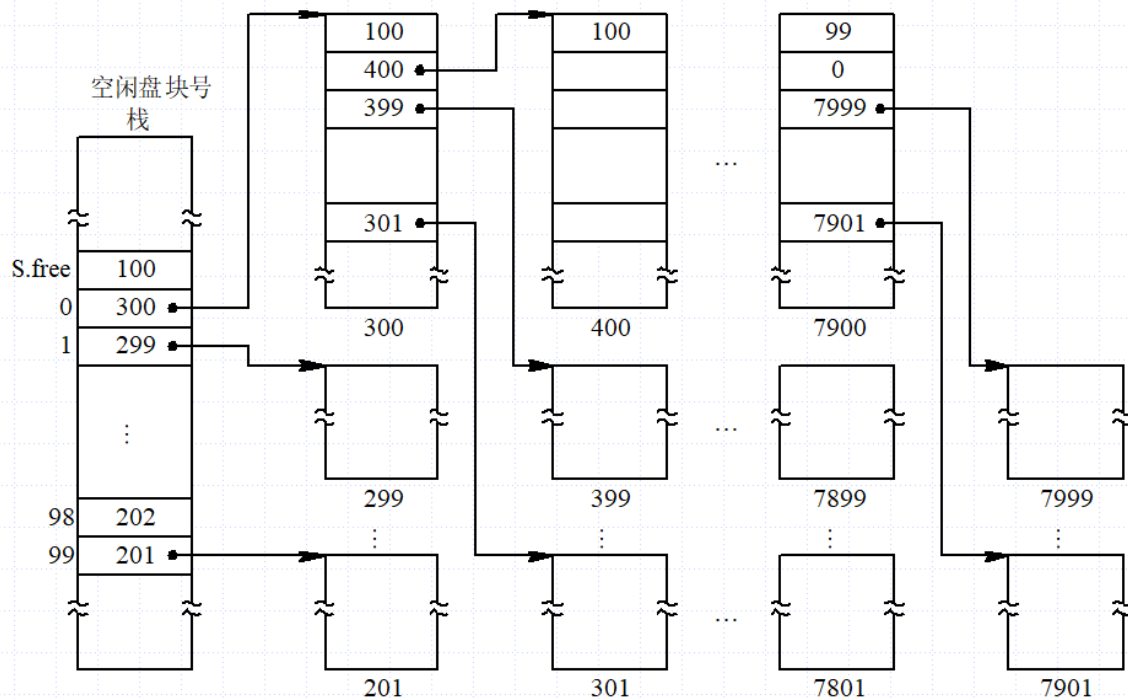
(2) 修改位示图。令 $\text{map}[i,j] = 0$ 。

例：如果利用20行，30列的位示图来标志空闲盘块的状态，在进行盘盘块分配时，当第一次找到的空闲盘块（即该位置为0）处于第11行，第18列，则相应的盘块号为_____。

$$b=n(i-1)+j=30(11-1)+18=318$$

成组链接法

成组链接法



一些点

FAT, 文件分配表 (名词)

作用

宽度, 以及它会影响啥东西

会计算 (272-273)

以上都是重点

主要掌握增量式索引组织方式 (! ! !) P277

位示图法 (重点)

第九章 操作系统接口

1、接口分类

联机命令接口
系统调用
图形用户接口

2、系统调用

在操作系统核心中设置了一组用于实现各种系统功能的子程序（过程），并将他们提供给应用程序调用。应用程序利用一种系统调用命令，去调用所需的系统过程。

系统调用是用户程序取得OS服务的唯一途径。

系统调用是一种特殊的过程调用。

系统态，用户态

特权指令，非特权指令